

Bibliothèques numériques I

Gestion de projet (conception, partage, archivage)

Markdown et LaTeX

Simon Gabay



Markdown

Markdown

Markdown est un langage de balisage léger créé en 2004.

Il existe un tutoriel pour les spécialistes de sciences humaines: Sarah Simpkin, « Débuter avec Markdown », traduction par Sofia Papastamkou, *Programming Historian en français* 2 (2020), <https://doi.org/10.46430/phfr0007>.

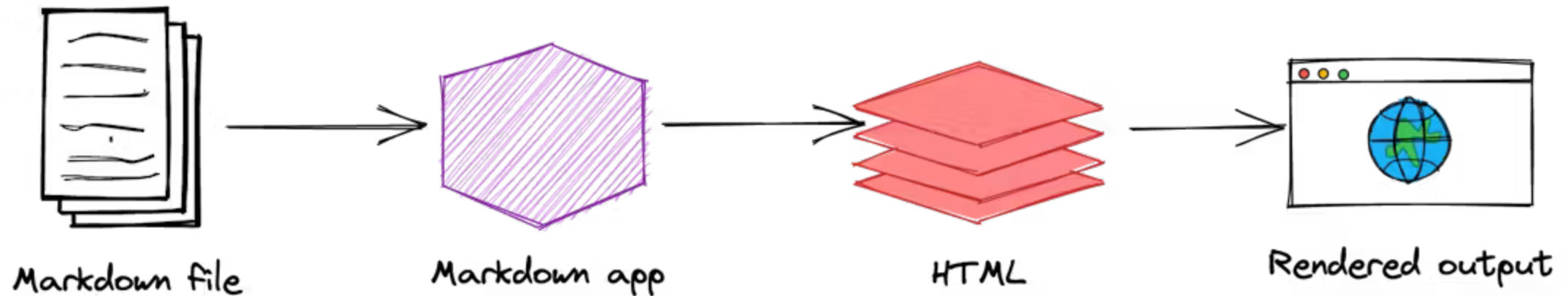
Exemple

On utilise des balises et une syntaxe très simple pour encoder des informations:

Code Markdown	Rendu
du **gras** ici	du gras ici
de l' <i>_italique_</i> ici	de l' <i>italique</i> ici
Un [lien vers Google](https://www.google.ch)	Un lien vers Google

Compiler

Pour obtenir le résultat final, on convertit par exemple le markdown en HTML (mais pas uniquement). On dit que l'on "compile" le code (source) en autre chose.



(Originellement, la compilation concerne le passage du code source en code objet. Tous les langages ne sont pas compilés, certains sont interprétés, comme Python ou R, d'autres sont compilés, comme Java).

Utiliser Markdown

Pour utiliser Markdown:

- De nombreux éditeurs proposent un compilateur intégré (comme Oxygen, Visual Studio Code...)
- Il existe des applications en ligne comme <https://markdownlivepreview.com>
- Pour convertir des documents en ligne de commande, vous avez [Pandoc](#)

Exercice

- Markdown est notamment utilisé avec GitLab/Github. Allez regarder comment cela se passe:
 - Aller sur le site du cours: <https://github.com/gabays/32M7128> ;
 - Sélectionner le `README.md` ;
 - Cliquer sur `Raw` en haut à droite pour voir le code non compilé.
- Tentez de rédiger un document en vous aidant de la [cheatsheet](#).

BibTeX

Lecture

Si vous voulez vraiment bien comprendre `LaTeX` avec une approche de spécialiste de sciences humaines, il existe un excellent ouvrage rédigé par Maïeul Rouquette, un des meilleurs spécialistes de la question:

```
@book{rouquette:halshs-00924546,  
  TITLE = {{{Xe)LaTeX appliqu{\`e} aux sciences humaines}},  
  AUTHOR = {Rouquette, Ma{\`i}eul},  
  URL = {https://halshs.archives-ouvertes.fr/halshs-00924546},  
  PUBLISHER = {{Atramenta}},  
  PAGES = {268},  
  YEAR = {2012},  
  KEYWORDS = {informatique ; bibliographique ; latex},  
  PDF = {https://halshs.archives-ouvertes.fr/halshs-00924546/file/latex-sciences-humaines.pdf},  
  HAL_ID = {halshs-00924546},  
  HAL_VERSION = {v1},  
}
```

Encoder sa bibliographie

Profitons de cet exemple pour introduire un nouveau langage: **BibTeX** -- ou plutôt un format conçu pour les bases bibliographiques.

```
@book{rouquette:halshs-00924546,  
  TITLE = {(Xe)LaTeX appliqu{\`e} aux sciences humaines},  
  AUTHOR = {Rouquette, Ma{"i}eul},  
  URL = {https://halshs.archives-ouvertes.fr/halshs-00924546},  
  PUBLISHER = {{Atramenta}},  
  YEAR = {2012},  
}
```

Pour que **LaTeX** puisse mettre en page votre bibliographie, il a besoin de savoir où se trouve le nom de l'auteur, le titre, la date de publication... **BibTeX** permet justement de formater toutes les données bibliographiques

Types d'entrée

On va distinguer:

- Les livres `@book{`
- Les articles `@article{`
- Les actes de conférence `@inproceedings{`
- Les rapports, les thèses, les billets de blogs, les logiciels...

suivi d'un code unique qui peut être signifiant (`duby_trois_ordres_1981`) ou pas (`gr1564freg45`) pour identifier l'entrée. Il est ainsi possible de "relier" le document LaTeX avec le fichier contenant les données bibliographiques lors de la compilation

S'il existe une multitude de types, tous les styles bibliographiques ne disposent pas de règle de mise en page pour chacun: il est possible que certains ne soient pas utilisables.

Les données

Il existe une multitude de champs

- Auteur(s): `author="Duby, Georges"`
- Titre: `title="Les Trois Ordres ou l'imaginaire du féodalisme"`
- Date de publication `year=1981`

Certains sont obligatoires pour certains types de publication

- Le nom de la revue et le numéro de volume pour les articles
- Maison d'édition pour les livres
- Université de soutenance pour une thèse
- ...

Ne pas oublier la virgule après chaque champ!

Quelques *tips*

- Pour les auteurs on utilise `Nom, Prénom` auivi de `and` entre chaque auteur s'il y en a plusieurs (`Duby, Georges and Le Goff, Jacques`)
- Si on veut conserver des majuscules en dépit du style pour la bibliographie on les met entre `{accolades}` (par ex. `title="{L}e {XVIe} siècle`)
- Le contenu peut être entre `"guillemets"` ou entre `{accolades}`
- Attention, parfois le nom des champs peut changer (`journal` vs `journaltitle` `date` vs `year` ...). Le compilateur vous indiquera les erreurs
- Parfois certains styles utilisent tous les champs renseignés, alors qu'ils peuvent faire doublon (par ex. url et DOI)

-> **Relisez toujours très soigneusement votre bibliographie!**

Gérer sa bibliographie

Vous pouvez tout encoder à la main (...) ou utiliser un logiciel de gestion de références comme [Zotero](#) qui génère un export en `BibTex` (gratuit) ou *EndNote* (payant) à partir de la base de données.

- Il est possible d'utiliser un *plug in* (ou *extension* en fonction de votre navigateur) pour sauvegarder des références depuis une page web (d'un catalogue de bibliothèque, ou du site d'une revue scientifique)
- Zotero est aussi compatible avec *Word* et *Libre office* pour mettre une bibliographie dans un éditeur de texte

LaTeX

Pourquoi LaTeX?

- LaTeX est entièrement gratuit
- LaTeX bénéficie d'un meilleur moteur de composition (gestion des espaces blancs)
- LaTeX permet une mise en page de bien meilleure qualité qu'un éditeur de texte classique, quasi professionnelle
- LaTeX est utilisé par de nombreuses conférences, qui exigent ce format (par exemple [CHR](#)) ou le recommandent (par ex. Humanistica)
- LaTeX peut être intégré dans une *pipeline* car c'est du code
- LaTeX ne permet pas d'éditeur que des articles, mais aussi des éditions ou tout autre document complexe (apparaît critique sur plusieurs niveaux, formules mathématiques...)

Tex

`TeX` est un système logiciel libre de composition de documents. `LaTeX` est un langage et un système de composition de documents qui permet de simplifier l'utilisation de `Tex`. Il existe plusieurs distributions:

- `MiKTeX` pour Windows
- `MacTeX` pour Mac OS
- `Tex Live` pour Mac OS

Ils doivent être installés sur votre machine pour utiliser `LaTeX` en local.

Les moteurs de composition

Il est nécessaire de compiler les documents `LaTeX` pour obtenir un pdf. Pour cela il existe plusieurs moteurs de composition:

- `pdfTeX` qui n'est plus développé (mais reste utilisable), et duquel dérive les deux suivants
- `XeTeX` modifie le moteur `TeX` pour élargir le nombre de caractères disponibles (encodage unicode+gestion des fontes OpenTypes)
- `LuaTeX` qui utilise Lua

Nous utiliserons `XeLaTeX`, qui permet d'utiliser des commandes `LaTeX` avec `XeTeX`.

Attention: tous ces moteurs ne sont pas compatibles, et un document écrit pour l'un ne fonctionnera pas avec l'autre!

Utiliser **LaTeX**

Deux options:

- Créer un compte sur [Overleaf](#)
- Installer en local
 - Télécharger la bonne distribution [en suivant ce guide](#)
 - Télécharger un éditeur, comme [TeXStudio](#).

Les bases

1. Que les commandes commencent par `\`
2. avec des arguments entre `{ accolades }`
3. des arguments optionnels entre `[crochets]`

```
\maketitle  
\textit{du texte en italique}  
\includegraphics[height=2cm]{images/logoLatex.png}
```

Un document simple

```
\documentclass{article}% la classe
\usepackage{fontspec}%gestion des caractères spéciaux (notamment les accents)

\begin{document}
Bonjour tout le monde!

C'est mon premier document LaTeX.
\end{document}
```

On observe la présence

1. D'un préambule, qui contient toutes les commandes qui affectent le document entier.
2. Du corps du document

Attention `\usepackage{fontspec}` est spécifique à `XeLaTeX`

Quelques premières commandes

1. `\textit` pour l'italique

```
\textit{texte à mettre en italique}
```

2. `\textbf` pour le gras

```
\textbf{texte à mettre en gras}
```

3. `\textsc` pour les petites capitales

```
\textsc{texte à mettre en petites capitales}
```

Commandes plus complexes

Dans certains cas, il n'est pas possible de tout mettre entre `{ accolades }` : il faut ouvrir et fermer l'élément (pour reprendre une terminologie XML):

```
\begin{itemize}  
  \item un  
  \item deux  
  \item trois  
\end{itemize}
```

Commandes encore plus complexes

Il est parfois utile d'appeler un fichier externe au document `LaTeX`, comme une image

```
\begin{figure}[!htb]
  %je centre l'image
  \centering
  %je précise la taille de mon image et le chemin vers elle
  \includegraphics[height=2cm]{images/logoLatex.png}
  %je rajoute une légende
  \caption{ici un logo}
  %je lui donne un identifiant pour pointer vers l'image par la suite
  \label{fig:logolatex}
\end{figure}
```


Exercices

Vous trouverez deux exercices dans le dossier:

- Un premier avec un document simple que vous devrez adapter à vos besoin
- Un document de conférence (ici Humanistica) pour écrire votre papier